WHAT IS CLAIMED IS:

1.   In a data processing system, a method for reordering a plurality of memory access requests, the method comprising steps of:

    accepting the plurality of requests;

    selecting a request to access an available memory location, from the plurality of requests; and

    scheduling the selected request.

2.   The method of claim 1, wherein the step of selecting a request comprises a step of selecting an oldest request, of the plurality of requests, that requests access to an available memory location.

3.   The method of claim 2, wherein the oldest request is oldest according to an order in which the plurality of requests were accepted in the accepting step.

4.   The method of claim 2, wherein the oldest request is oldest according to time fields associated with the plurality of requests.

5.   The method of claim 1,

    wherein the step of selecting a request comprises steps of:

    determining whether a read request to access an available memory location exists, among the plurality of requests;

    selecting a read request to access an available memory location, if the determining step determines that a read request exists; and

    selecting a non-read request to access an available memory location, if the determining step determines that a read request does not exist.

6.   The method of claim 5, wherein:

the step of selecting a read request comprises a step of selecting a read request to access an available memory location according to an original ordering of the plurality of operations; and

the step of selecting a non-read request comprises a step of selecting a non-read request to access an available memory location according to an original ordering of the plurality of operations.

7.   The method of claim 5 further comprising steps of:

determining whether the selected request is a write request; and

after the step of scheduling the selected request, successively scheduling at least one write request from the plurality of requests, if the selected request is determined to be a write request.

8.   The method of claim 7, wherein the step of successively scheduling at least one write request comprises steps of:

counting write requests that have been successively scheduled; and

successively scheduling write requests until a predefined maximum number of write requests has been counted in the counting step.

9.   The method of claim 7, wherein the step of successively scheduling at least one write request comprises steps of:

determining whether a write request upon an available memory location exists among the plurality of requests; and

stopping the scheduling of write requests if a write request upon an available memory location is determined not to exist.

10.   The method of claim 7, wherein the step of successively scheduling at least one write request comprises steps of:

counting write requests that have been successively scheduled;

determining whether a write request upon an available memory location exists among the plurality of requests; and

stopping scheduling of write requests upon one of the following occurrences:

a predefined maximum number of write requests has been counted in the counting step, and

a write request upon an available memory location is determined not to exist.

11.   The method of claim 7, wherein at least one request from the plurality of requests produces a return value, the method further comprising steps of:

collecting return values of requests of the plurality of requests; and

ordering the collected return values according to an original ordering of the plurality of requests.

12.   The method of claim 7, wherein the step of accepting the plurality of requests includes a step of dynamically accepting new memory access requests into the plurality of requests.

13.   The method of claim 5, wherein at least one request from the plurality of requests produces a return value, the method further comprising steps of:

collecting return values of requests of the plurality of requests; and

ordering the collected return values according to an original ordering of the plurality of requests.

14.   The method of claim 5, wherein the step of accepting the plurality of requests includes a step of dynamically

accepting new memory access requests into the plurality of requests.

15.   The method of claim 5, wherein the step of selecting a non-read request comprises a step of determining whether a write path is clear.

16.   The method of claim 1, wherein at least one request from the plurality of requests produces a return value, the method further comprising steps of:

collecting return values of requests of the plurality of requests; and

ordering the collected return values according to an original ordering of the plurality of requests.

17.   The method of claim 1, wherein the step of accepting the plurality of requests comprises a step of dynamically accepting a new memory access request into the plurality of requests.

18.   The method of claim 1 further comprising a step of detecting whether the plurality of requests contains a read request and a write request to a same memory location.

19.   The method of claim 18, wherein the ensuring step comprises steps of:

determining whether a write request exists in the plurality of requests that requests access to a same memory location to which access is requested by a new read request;

if it is determined that the write request to the same location does not exist, adding the new read request to the plurality of requests; and

if it is determined that the write request to the same location does exist, delaying adding the new read request to the plurality of requests until the write request to the same location has been scheduled and is no longer in the plurality of requests.

20.   The method of claim 1,

        wherein the step of accepting the plurality of requests comprises a step of accepting data associated with the plurality of requests; and

        the step of scheduling the selected request comprises a step of submitting data associated with the selected request.

21.   The method of claim 1, wherein each memory location is associated with one of a plurality of memory banks, and wherein the step of selecting a request comprises steps of:

        identifying a memory location to which access is requested by a request from the plurality of requests;

        identifying a memory bank, of the plurality of memory banks, to which the identified memory location corresponds; and

        determining whether the identified memory bank is available.

22.   The method of claim 1, wherein a module that generates requests of the plurality of requests is not aware of the order of scheduling chosen by the method.

23.   The method of claim 1, wherein the step of scheduling the selected request comprises a step of sending the selected request to a memory system for execution.

24.   The method of claim 1, wherein the step of scheduling the selected request comprises a step of placing the selected request into a queue.

25.   The method of claim 1, wherein the plurality of requests are generated by at least two processors.

26.   The method of claim 1, wherein the data processing system includes at least one Pentium Pro processor.

27.   The method of claim 1, wherein the data processing system includes at least two P6 buses.

28.   In a data processing system, a method for scheduling a plurality of requested memory operations, wherein the requested operations have default priorities, the method comprising steps of:

accepting the plurality of operations;

determining whether a particular operation of the plurality of operations operates upon a non-available memory location;

if the particular operation operates upon a non-available memory location, selecting a requested operation that operates upon an available memory location and has lower default priority than the particular operation, from the plurality of operations; and

scheduling the selected operation.

29.   The method of claim 28, wherein the step of selecting an operation comprises a step of selecting an operation with highest default priority among operations that operate upon available memory locations, from the plurality of operations.

30.   The method of claim 29, wherein a requested operation's default priority is determined by its age, wherein any one operation has higher default priority than any operation newer than the one operation.

31.   The method of claim 28, wherein each requested operation is of a type from a set of types including a more-critical type, and wherein the step of selecting an operation comprises steps of:

determining whether a more-critical operation exists, among the plurality of operations, that operates upon an available memory location;

selecting a more-critical operation that operates upon an available memory location, if the

determining step determines that a more-critical operation exists; and

        selecting an operation not of the more-critical type that operates upon an available memory location, if the determining step determines that a more-critical operation does not exist.

32.   The method of claim 31, wherein the more-critical type includes a read type.

33.   The method of claim 31, wherein the set of types includes a burstable type, the method further comprising steps of:

        determining whether the selected operation is a burstable operation; and

        after the step of scheduling the selected request, successively scheduling at least one burstable operation from the plurality of operations, if the selected operation is determined to be a burstable operation.
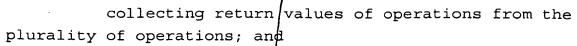
34.   The method of claim 33, wherein the burstable type includes a write type.

35.   The method of claim 28, wherein the set of types includes a burstable type, the method further comprising steps of:

        determining whether the selected operation is a burstable operation; and

        after the step of scheduling the selected request, successively scheduling at least one burstable operation from the plurality of operations, if the selected operation is determined to be a burstable operation.

36.   The method of claim 28, wherein at least one operation from the plurality of operations produces a return value, and wherein the plurality of operations have an original ordering, the method further comprising steps of:

collecting return values of operations from the plurality of operations; and

ordering the collected return values according to the original ordering.

37.   The method of claim 28, wherein the plurality of operations are generated by at least two processors.

38.   The method of claim 28, wherein a module that generates requested operations of the plurality of operations is not aware of the order of scheduling chosen by the method.

39.   In a data processing system, a method for ordering a plurality of requested memory operations, the method comprising steps of:

accepting the plurality of operations;

determining whether a particular operation of the plurality of operations operates upon a non-available memory location, and if so, selecting a requested operation newer than the particular operation that operates upon an available memory location from the plurality of operations, the step of selecting a requested operation comprising steps of:

determining whether a read operation exists, in the plurality of operations, that operates upon an available memory location;

if the read operation exists, selecting an oldest operation of read operations that operate upon available memory locations; and

if a read operation does not exist that operates upon an available memory location, selecting an oldest write operation, of the plurality of operations, that operates upon an available memory location; and

scheduling the selected operation.

40.   The method of claim 39, wherein the step of accepting the plurality of operations includes a step of dynamically accepting new requested memory operations into the plurality of operations.

41.   The method of claim 40, wherein at least one
operation from the plurality of operations produces a return
value, the method further comprising steps of:

collecting return values of operations from the
plurality of operations; and

ordering the collected return values according to
the age of their corresponding operations.

42.   The method of claim 41, further comprising, after
the scheduling step, additional steps of determining whether
the selected operation is a write operation, and if it is,
successively scheduling a burst of at least one write
operation from the plurality of operations.

43.   The method of claim 42, wherein each memory location
corresponds to one of a plurality of semiconductor memory
banks, and wherein the step of selecting a requested operation
comprises steps of:

identifying a memory location upon which an
operation from the plurality of operations operates;
identifying a memory bank, of the plurality of
memory banks, to which the identified memory location
corresponds; and

determining whether the identified memory bank
is available.

44.   In a computing system, a method for assigning
priority to memory access requests, wherein an oldest request
among the requests has a currently non-available target memory
location, the method comprising steps of:

determining current availability of target memory
locations of said memory requests; and

assigning high priority to memory access requests
having currently available target memory locations.

45.   The method of claim 44, wherein the step of
assigning high priority comprises steps of:

determining relative ages of requests having currently available target locations; and

assigning higher priority to older requests having currently available target locations than to newer requests having currently available target locations.

46.   The method of claim 44,

wherein the step of assigning high priority further comprises a step of determining which of the requests having currently available target locations are read requests; and

wherein the step of assigning higher priority assigning higher priority to read requests having currently available target locations than to write requests having currently available target locations.

47.   The method of claim 46, further comprising a step of scheduling the memory access requests according to their assigned priority.

48.   The method of claim 47, further comprising, after the scheduling step, steps of:

receiving return values of the requests; and

ordering the return values according to the age of their corresponding requests.

49.   In a data processing system, a method for scheduling a plurality of requested memory operations including read operations and write operations, wherein at any time, there exists a current preference that is one of a read preference, a write preference, and an indeterminate preference, the method comprising steps of:

determining the current preference;

determining whether there exists a write operation upon an available memory location among the plurality of operations;

determining whether there exists a read operation upon an available memory location among the plurality of operations;

scheduling a read operation upon an available memory location upon one of the following conditions:

it is determined in the determining steps that a read operation upon an available location exists, and that the current preference is read preference; and

it is determined in the determining steps that a read operation upon an available location exists, that no write operation upon an available location exists, and that the current preference is write preference.

50.   The method of claim 49, further comprising a step of scheduling a write operation upon an available memory location upon one of the following conditions:

it is determined in the determining steps that a write operation upon an available location exists, and that the current preference is write preference; and

it is determined in the determining steps that a write operation upon an available location exists, that no read operation upon an available location exists, and that the current preference is read preference; and

setting a new current preference after one of the steps of scheduling a read operation and scheduling a write operation.

51.   The method of claim 50, wherein the step of determining whether there exists a write operation upon an available memory location comprises a step of determining whether a write path is clear.

52.   The method of claim 50, wherein the plurality of requested memory operations have default priorities, and wherein:

the step of scheduling a write operation comprises a step of scheduling an operation with highest default priority among write operations upon available memory locations; and

the step of scheduling a read operation comprises a step of scheduling an operation with highest priority among read operations upon available memory locations.